THE INVENTION CLAIMED IS:

1. A sequencer which executes instructions based on a function clock signal to perform I/O functions in a serial peripheral interface based on a source clock signal, the function clock signal having a frequency of two times the frequency of the source clock signal, comprising:

an instruction store containing instructions at addresses;

a program counter connected to the instruction store and receptive of the function clock signal to create address signals and increment the address signals to address the instructions in the instruction store;

an instruction decoder connected to the instruction store for decoding the instructions to perform I/O transfers of bit signals at the source clock frequency; and

a function clock generator for generating the function clock signal, the function clock generator connected to the program counter and operative to cause the program counter to address a predetermined instruction decoded by the instruction decoder to selectively force the frequency of the function clock signal to be equal to the frequency of the source clock signal during the duration of that function clock signal during which the predetermined instruction is decoded.

2. A sequencer as defined in claim 1 wherein the predetermined instruction functionally causes the transfer of a single bit signal to or from the interface.

3. A sequencer as defined in claim 1 wherein the predetermined instruction further forces the frequency of the function clock signal to be equal to the frequency of the source clock signal for a predetermined number of sequential executions of that predetermined instruction.

4. A sequencer as defined in claim 3 further comprising:

a repeat counter connected to the instruction decoder to repeat the execution of the predetermined instruction once during a predetermined number of sequential periods of the clock source signal.

5. A sequencer as defined in claim 4 further comprising:

data path elements connected to the instruction decoder and responsive to instructions from the instruction decoder to transfer the bit signals from the multibit word to and from the interface;

the data path elements including a serialization register receptive of the multibit word to be transferred and a data out multiplexer connected to the serialization register; and wherein:

the data out multiplexer is selectively controllable by an instruction from the instruction decoder to select an output path from a bit position within the serialization register through the data output multiplexer.

6. A sequencer as defined in claim 5 wherein:

the data out multiplexer is connected to supply a bit signal in the output path from the serialization register to an output conductor of the interface upon the multiplexer selecting the output path.

7. A sequencer as defined in claim 5 wherein:

the data path elements further include a bit counter connected to the data out multiplexer for establishing the selected output path from the bit position within the serialization register through the data output multiplexer, an instruction from the instruction decoder establishing a count within the bit counter by which to index through all of the bit positions of the multibit word.

8. A sequencer as defined in claim 7 wherein the number of repetitions established by the repetition counter and the number of indexes established by the bit counter are coordinated to transfer all of the bits of the multibit word.

9. A sequencer as defined in claim 4 wherein:

data path elements connected to the instruction decoder and responsive to instructions from the instruction decoder to transfer the bit signals from multibit words to and from the interface;

the data path elements including a serialization register receptive of the multibit word to be transferred and a data path multiplexer connected to supply input signals to the serialization register in a feedback path leading from an output

terminal of the serialization register through the data path multiplexer to an input terminal of the serialization register.

10. A sequencer as defined in claim 9 wherein the data path elements further comprise:

merge logic connected in the feedback path between the serialization register and and the data path multiplexer;

a bit counter connected to the instruction decoder and operative to supply a bit position signal related to the bit position of a bit signal transferred to or from the interface;

a bit position decoder receptive of the bit position signal and operative to supply a logical signal to the merge logic related to the bit position signal; and wherein:

the merge logic logically determines each previous bit signal which has been transferred into or from the interface and updates those bit signals in the serialization register during the transfer of the bit signals to and from the interface.

11. A sequencer as defined in claim 1 wherein the predetermined instruction is a delay instruction which forces the frequency of the function clock signal to be equal to the frequency of the source clock signal for a predetermined number of sequential executions of that predetermined delay instruction.

12. A method of executing instructions based on a function clock signal to perform I/O functions in a serial peripheral interface based on a source clock signal when the function clock signal has a frequency of two times the frequency of the source clock signal, comprising the steps of:

locating instructions at addresses;

incrementing addresses to select instructions at the rate of one instruction per cycle of the function clock signal;

decoding the selected instructions to perform I/O transfers of its signals at the source clock frequency; and

selectively forcing the frequency of the function clock signal to be equal to the frequency of the source clock signal as the result of decoding a

predetermined instruction executed during the period of the function clock signal during which the predetermined instruction is decoded.

13. A method as defined in claim 12 further comprising the step of:

causing the transfer of a single bit signal to or from the interface as a result of executing the predetermined instruction.

14. A method as defined in claim 12 further comprising the step of:

forcing the frequency of the function clock signal to be equal to the frequency of the source clock signal for a predetermined number of sequential executions of the predetermined instruction.

15. A method as defined in claim 14 further comprising the step of:

repeating the execution of the predetermined instruction once during a predetermined number of sequential periods of the clock source signal.

16. A method as defined in claim 12 further comprising the step of:

selecting the predetermined instruction as a delay instruction; and

causing a delay in the execution of other instructions by forcing the frequency of the function clock signal to be equal to the frequency of the source clock signal for a predetermined number of sequential executions of predetermined delay instruction as a result of decoding the delay instruction.